

C 語言概述

1.1 C語言的發展簡史和特點

1. C語言的誕生與發展

(1) 在C語言誕生以前，系統軟體主要是用組合語言編寫的。由於組合語言程式依賴於電腦硬體，其可讀性和可移植性都很差；但一般的高級語言又難以實現對電腦硬體的直接操作（這正是組合語言的優勢），於是人們盼望有一種兼有組合語言和高級語言特性的新語言。

(2) C語言是貝爾實驗室於70年代初研製出來的，後來又被多次改進，並出現了多種版本。80年代初，美國國家標準化協會（ANSI），根據C語言問世以來各種版本對C語言的發展和擴充，制定了ANSI C標準（1989年再次做了修訂）。

本書以ANSI C新標準來介紹。

(3) 目前，在微機上廣泛使用的 C 語言編譯系統有 Microsoft C、Turbo C、Borland C 等。雖然它們的基本部分都是相同的，但還是有一些差異，所以請大家注意自己所使用的 C 編譯系統的特點和規定（參閱相應的手冊）。

本書選定的上機環境是 *TC V2.0 (DOS 操作系統)*。

2. C 語言的特點

C 語言同時具有組合語言和高級語言的優勢。

- (1) 語言簡潔、緊湊，使用方便、靈活。
- (2) 運算符極其豐富。
- (3) 生成的目標代碼品質高，程式執行效率高。
- (4) 可移植性好（較之組合語言）。
- (5) 可以直接操縱硬體。

3. 在C語言中，除實現**順序**、**選擇**和**迴圈**三種基本結構等的9條控制語句外，輸入輸出操作均由標準庫函數（不是C語言的組成部分）來實現。

所以**學習C語言**，不僅要學習這9條控制語句和各種運算符，而且要學習並掌握常用標準庫函數的使用。

[Return]

1.2 C 語言程式的結構與書寫規則

1.2.1 C 語言程式的總體結構

一個完整的 C 語言程式，是由一個main()函數（又稱主函數）和若干個其他函數結合而成的，或僅由一個main()函數構成。

[**案例1.1**] 僅由main()函數構成的 C 語言程式。

```
/*案例代碼檔案名：AL1_1.C*/
```

```
/*功能：僅由main()函數構成的 C 語言程式示例*/
```

```
main()
```

```
{ printf(“This is a C program.\n”);
```

```
}
```

程式運行結果：

This is a C program.

[[程式演示](#)]

[**案例1.2**] 由main()函數和1個其他函數max()構成的 C 語言程式。

```
/*案例代碼檔案名： AL1_2.C*/
```

```
/*功能： 由main()函數和1個其他函數max()構成的 C 語言程式示例*/
```

```
int max(int x, int y)
```

```
{ return( x>y ? x : y ); }
```

```
main()
```

```
{ int num1,num2;
```

```
printf(“Input the first integer number: ”);
```

```
scanf(“%d”, &num1);
```

```
printf(“Input the second integer number: ”);
```

```
scanf(“%d”, &num2);
```

```
printf(“max=%d\n”, max(num1, num2));
```

```
}
```

程式運行情況：

```
Input the first integer number:6←┘
```

```
Input the second integer number:9←┘
```

```
max=9
```

[[程式演示](#)]

[**案例1.3**] 改寫[**案例1.2**]，交換main()函數和max()函數的前後位置。
根源程式略。

程式運行情況：

Input the first integer number:6←┘

Input the second integer number:9←┘

max=9

1.函數是 C 語言程式的基本單位。

main()函數的作用，相當於其他高級語言中的主程序；其他函數的作用，相當於副程式。

2.C語言程式總是從main()函數開始執行。

一個 C 語言程式，總是從main()函數開始執行，而不論其在程式中的位置。當主函數執行完畢時，亦即程式執行完畢。

習慣上，將主函數main()放在最前頭。

1.2.2 函數的一般結構

任何函數（包括主函數main()）都是由函數說明和函數體兩部分組成。其一般結構如下：

[函數類型] 函數名(函數參數表)	}	函數說明部分
{ 說明語句部分;	}	函數體部分
執行語句部分;		
}		

1. 使用的語法符號約定

[...]——方括號表示可選（即可以指定，也可以缺省）

.....——省略號表示前面的項可以重複

| ——多（含2）中選1

2. 函數說明

由函數類型（可缺省）、函數名和函數參數表三部分組成，其中函數參數表的格式為：

數據類型 形參[, 數據類型 形參2.....]

例如，[案例1.2]中的函數max()，其函數說明各部分如圖1-1所示。

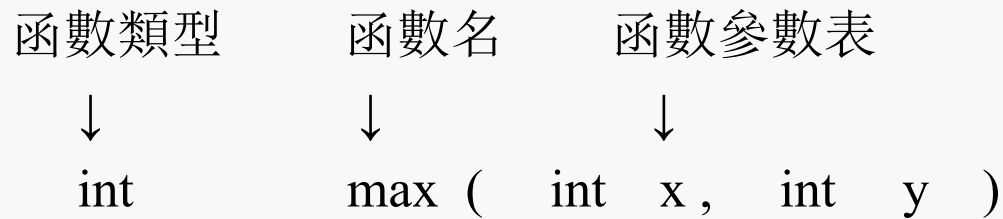


圖1-1 函數說明部分結構圖

注意：在舊標準中，函數可以缺省參數表。

3.函數體

在函數說明部分的下麵、大括弧（必須配對使用）內的部分。

函數體一般由說明語句和可執行語句兩部分構成：

（1）說明語句部分

說明語句部分由變數定義、自定義類型定義、自定義函數說明、外部變數說明等組成。

(2) 可執行語句

一般由若干條可執行語句構成。圖1-2是[**案例1.2**]的main()函數體的示意圖。

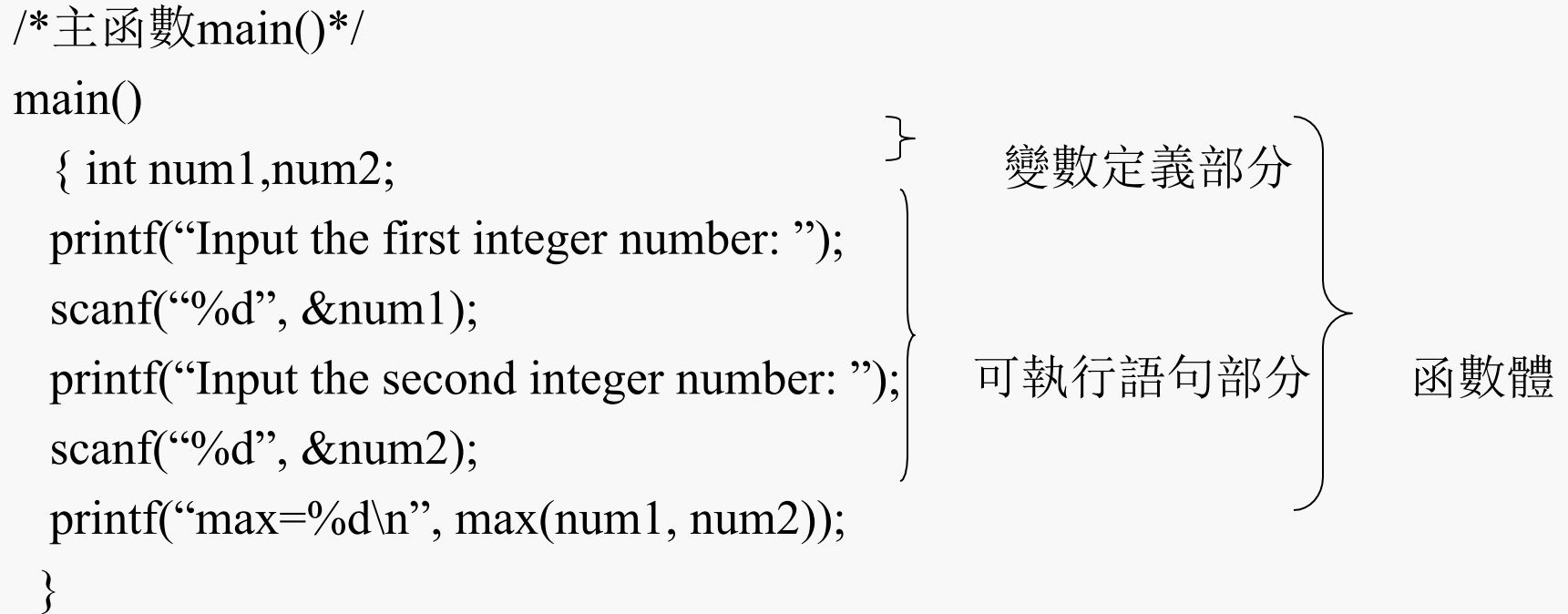


圖1-2 函數體結構示意圖

4.說明

(1) 函數體中的變數定義語句，必須在所有可執行語句之前。
下麵程式中變數定義語句“int max; ”的位置是非法的：

```

main()
{ int x,y;          /*變數定義語句：定義2個整型變數x、y*/
  x=3;             /*可執行的賦值語句：將3賦值給變數x*/
  y=6;             /*可執行的賦值語句：將6賦值給變數y*/
  int max;         /*變數定義語句：出現在可執行的
                   賦值語句“x=3;”和“ y=6;”之後，非法！*/
  max=x>y?x:y;
  printf(“max=%d\n”,max);
}

```

解決辦法很簡單，請讀者自己思考。

(2) 如果不需要，也可以缺省變數定義語句。

1.2.3 根源程式書寫格式

1.所有語句都必須以分號“;”結束，函数的最後一個語句也不例外。

2.程式行的書寫格式自由，既允許1行內寫幾條語句，也允許1條語句分寫在幾行上。

例如，[**案例1.2**]的主函數main()，也可改寫成如下所示的格式：

.....

```
main()
{ int num1,num2;
  printf("Input the first integer number: "); scanf("%d", &num1);
  printf("Input the second integer number: "); scanf("%d", &num2);
  printf("max=%d\n", max(num1, num2));
}
```

如果某條語句很長，一般需要將其分寫在幾行上。

3.允許使用注釋。

C語言的注釋格式為： /* */

例如，在[**案例1.1**]和[**案例1.2**]中，以及本節其他部分給出的根源程式中，凡是用“/*”和“*/”括起來的文字，都是注釋。

(1) “/*”和“*/”必須成對使用，且“/”和“*”、以及“*”和“/”之間不能有空格，否則都出錯。

技巧：為避免遺漏必須配對使用的符號，例如注釋符號、函數體的起止識別字（花括弧）、圓括號等等，在輸入時，可連續輸入這些起止識別字，然後再在其中進行插入來完成內容的編輯。在起止識別字嵌套時，以及相距較遠時，這樣做更有必要。

(2) 注釋的位置，可以單占1行，也可以跟在語句的後面。

(3) 如果1行寫不下，可另起1行繼續寫。

(4) 注釋中允許使用漢字。在非中文操作系統下，看到的是一串亂碼，但不影響程式運行。

[\[Return\]](#)

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/106103000105010220>